Real Time Freshness Detection for Mint Plant using Edge Computing Device

Priyadharshini Ganesan, Kanaga Shanmugi
Dept. Communication Networks
University of Bremen, Germany
Corresponding author: pganesan@uni-bremen.de, kans@uni-bremen.de

Abstract

This paper presents the design and implementation of an embedded system for real-time freshness classification using the ESP32S3-EYE microcontroller, combining edge computing with machine learning. The system classifies perishable items, in this case Mint leaves into three categories: Fresh, Dried, and Spoiled based on image captured through the onboard camera. A custom-trained convolutional neural network (CNN) model is deployed using TensorFlow Lite Micro, optimized for inference on resource-constrained hardware. The system leverages the ESP32S3-EYE's camera for image acquisition and its low-power processing capabilities to perform on-device classification without requiring cloud connectivity. The proposed system offers a low-cost, energy-efficient solution suitable for IoT applications, including smart kitchens, supply chain monitoring, and food quality management, demonstrating a viable approach to enhance food safety and reduce waste in real-world settings.

Keywords—ESP32S3-EYE, Edge AI, Freshness Classification, Food Quality, Convolutional Neural Networks (CNN), Real-Time Processing, Low-Power Computing.

1 Introduction and Motivation

The proliferation of Internet of Things (IoT)devices has driven advancements in edge computing, allowing it for real-time processing and decision-making. In particular, machine learning (ML) on edge devices has become increasingly important in applications such as health monitoring, environmental sensing, and food quality control. Ensuring the freshness of perishable goods, such as fruits, vegetables, and other consumables, is a critical aspect of supply chain management and household consumption [5]. Traditionally, food freshness classification has relied on subjective human judgment or centralized, cloud-based systems, which are limited by network reliability, latency, and power consumption.

The main motivation of the project is to avoid food waste and improve the yield. Food waste is a critical global issue, contributing to both economic loss and environmental degradation. According to the Food and Agriculture Organization (FAO), approximately one-third of all food produced globally is wasted, amounting to about 1.3 billion tons per year. This wastage occurs at various stages of the supply chain, from farm to the dinning table. Tackling this problem not only involves improving production efficiency but also enhancing monitoring systems to better manage food freshness and reduce unnecessary waste. The challenge of food waste intersects with several of the United Nations' Sustainable Development Goals (SDGs), particularly Goal 2: Zero Hunger, Goal 12: Responsible Consumption and Production, and Goal 13: Climate Action[2].

The proposed project seeks to address these issues by leveraging edge computing and embedded machine learning for real-time food freshness classification. By automating the process of determining whether food is Fresh, Dried, or Spoiled, it aim to provide a cost-effective, energy-efficient solution that contributes to the sustainable management of food resources. This paper present a compact, cost-effective, and energy-efficient freshness classification system using the ESP32S3-EYE microcontroller combined with TensorFlow Lite Micro to deploy a custom-trained convolutional neural network (CNN). This enables the system to make quick decisions with minimal power consumption, a critical advantage in low-resource environments or applications with limited connectivity.

2 Hardware Setup

The hardware setup Figure 1 involves the ESP32S3-EYE, with an integrated camera and display, used for real-time freshness prediction. The ESP32S3-EYE captures images of the mint plant and runs a pre-trained CNN model on-device to classify the plant's freshness as Fresh, Dried, or Spoiled. The prediction result is displayed on the ESP32S3-EYE's screen and serial port.

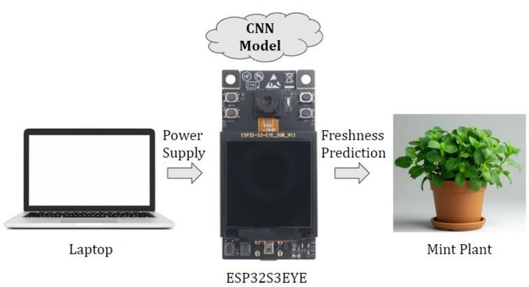


Figure 1: Setup Block Diagram

The laptop provides the power supply via a USB connection and can be used during development for tasks such as loading the CNN model onto the device, debugging, and monitoring system performance. The entire setup allows for edge-based AI processing, eliminating the need for cloud computation, making it efficient for real-time freshness detection. The protype of the setup used for demonstration is shown in Figure 2

Figure 2: Hardware Setup

2.1 Deep Learning based method

2.1.1 Methodology

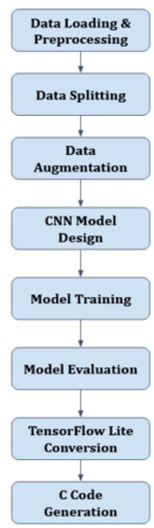


Figure 3: Methodology for Training the model

The deep learning-based method for freshness classification follows the structured workflow outlined in Figure 3. The steps are as follows:

Data Loading and Preprocessing: The images undergo preprocessing operations, including resizing and normalization, to ensure uniformity across the dataset, preparing them for model training as explained in [4].

Data Splitting: The dataset is divided into training, validation, and testing subsets.In this project, 20% of the dataset is used for both Testing and validation together.

Data Augmentation: Data augmentation techniques such as rotation, flipping, and scaling are applied to artificially increase the dataset size and diversity. The image size used is 96 x 96 with 3 channels for RGB.

CNN Model Design: A custom Convolutional Neural Network (CNN) is designed for the task of freshness classification. The model is structured with convolutional layers for feature extraction, followed by fully connected layers for classification into the categories of Fresh, Dried, or Spoiled. The custom Convolutional Neural Network (CNN) used in this project is explained in detail in **Section 2.1.3**.

Model Training: The CNN is trained using the training dataset.

Model Evaluation: After training, the model is evaluated on validation and testing datasets. Evaluation metrics such as accuracy and precision are computed to assess the performance of the model on unseen data.

TensorFlow Lite Conversion: Once the model has been trained and evaluated, it is converted into TensorFlow Lite format, enabling it to run efficiently on embedded devices such as the ESP32S3-EYE.

C Code Generation: The TensorFlow Lite model is integrated into C code, allowing it to be deployed on the ESP32S3-EYE for real-time inference.

2.1.2 Implementation

Figure 4 depicts the workflow for implementing the Machine Learning model on the ESP32S3-EYE module for real-time freshness prediction. The key steps are as follows:

Image Capture: The ESP32S3-EYE, equipped with an onboard camera, captures real-time images of the object to be classified i.e, Mint Leaves and the Unknown datasets. This image serves as input for the classification process and is captured in RGB format.

Processing with CNN Model: The captured image is passed through the deployed CNN model on the ESP32S3-EYE. The model processes the image and classifies the object based on visual features into one of the three freshness categories: Fresh, Dried, or Spoiled.

Freshness Detection: The classification results are computed based on the model's output. The system determines whether the object is fresh, dried, or spoiled.

Display of Results: The classification result is displayed on the ESP32S3-EYE's integrated screen and the serial port.

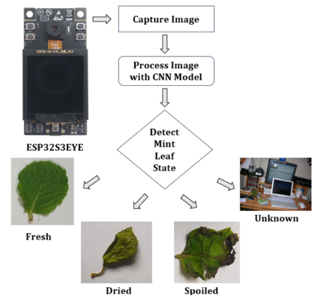


Figure 4: Implementation on the ESP32 S3 EYE module

2.1.3 Model Architecture

The model architecture used for freshness classification is based on a Convolutional Neural Network (CNN), a widely adopted deep learning architecture for image classification tasks [1]. Figure 5 illustrates the general structure of the CNN used for this project.

Input Layer: The model takes an image of the object, such as a leaf, as the input. The image is represented as a matrix of pixel values, which is processed to extract important visual features related to freshness.

Convolutional Layers: The image is passed through multiple convolutional layers. In these layers, filters are applied to detect various features such as edges, textures, and patterns that are indicative of the freshness state. Each filter generates a feature map that highlights specific visual aspects of the input. (Input Layer- 8, Hidden Layer 1-16, Hidden Layer 2-32, Dense Layer 1-32, Dense Layer 1-32 neurons).

Pooling Layers: After each convolution operation, pooling layers are used to reduce the dimensionality of the feature maps, retaining the most important information while discarding irrelevant details. This also helps to reduce computational complexity and prevent over fitting.

Feature Maps: The feature maps generated by the convolutional and pooling layers represent progressively higher-level abstractions of the input image, such as color changes, decay patterns, or structural differences between fresh, dried, and spoiled items.

Fully Connected Layers: The output from the final convolutional and pooling layers is flattened into a vector and fed into a series of fully connected (dense) layers. These layers combine the features extracted by the CNN and perform the classification task.

Output Layer: The final output is a softmax layer, which provides the probability distribution over the possible freshness categories: Fresh, Dried, Spoiled, and Unknown. The category with the highest probability is chosen as the model's prediction.

The architecture efficiently handles the image classification task by hierarchically extracting and combining relevant visual features. It is optimized for deployment on low-power, resource-constrained devices like the ESP32S3-EYE.

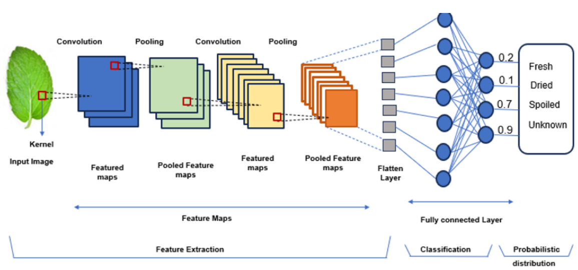


Figure 5: Model architecture of a Typical Convolution Layer

2.1.4 Dataset

Table 1 provides an overview of the dataset used for training, testing, and validating the freshness classification model [3]. The dataset can be accessed through [1]. The dataset consists of four categories: Fresh, Dried, Spoiled, and Unknown.

Table 1: Dataset

	Fresh	**Dried**	**Spoiled**	**Unknown**
Training	1699	1699	1699	1699
Testing	170	170	170	170
Validation	170	170	170	170

[1]Download the Freshness Classification dataset

2.1.5 Training and Evaluation Results

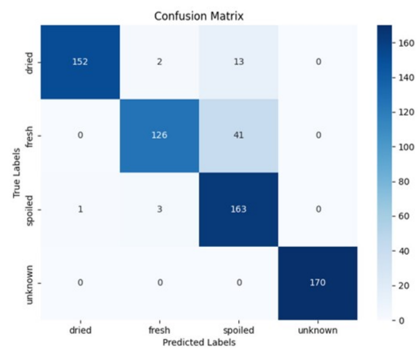


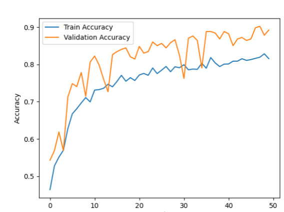
Figure 6: Confusion Matrix

Figure 6 shows the confusion matrix for the model's classification results on the testing dataset. The confusion matrix provides a detailed breakdown of the model's performance across all four categories: Fresh, Dried, Spoiled, and Unknown. The model correctly classified 152 Fresh images, 126 Dried images, 153 Spoiled images, and 170 Unknown images.

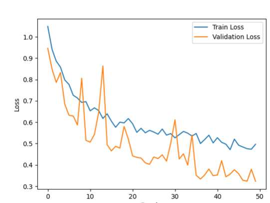
Because, the classification is aimed to detect freshness level between mint leaves other image criterias when detected should be shown as Unknown. So the 170 out of 170 images detected were correctly classified as Unknown.

Clearly some of the Dried and spoiled samples were falsely detected i.e., false positive and false negative, this is because the images used in dataset is not having enough variation to differentiate between spoiled and dried category.

2.1.6 Model Training Metrics



a) Accuracy plot Train vs. Validation



b) Loss plot Train vs. Validation

Figure 7: Monitor the output on LCD for new model

Figure 7 illustrates the training and validation accuracy and loss curves over epochs during model training. Figure 7 a) The graph shows the accuracy trends for both the training and validation sets over the course of 50 epochs. The training accuracy increases steadily as the model learns from the data, reaching approximately 92%. The validation accuracy shows fluctuations but generally improves over time, indicating the model's ability to generalize to unseen data. Figure 7 b) plot shows the training loss and validation loss, where the loss represents the error between the predicted outputs and the actual labels. The training loss decreases consistently, indicating that the model is improving its predictions. The validation loss shows a more fluctuating trend but stabilizes toward the end, demonstrating that the model generalizes well without significant over fitting or under fitting.

2.1.7 Performance Evaluation

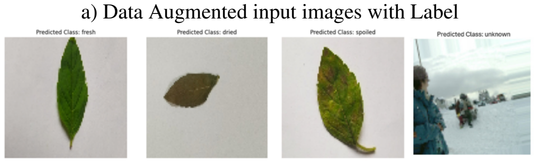
Table 2: Performance Evaluation

Metrics	Value
Accuracy	0.9105
Precision	0.9269
Recall	0.9105
F1 Score	0.9115

Table 2 presents the performance evaluation metrics for the trained model on the testing dataset.

Accuracy: The model achieves an accuracy of approximately 91%, which indicates a high overall performance in classifying the images into the correct freshness categories.

a) Data Augmented input images with Label



b) Sample Predictions using the generated Model

Figure 8: Image Pre-processing & Sample output

Precision: A precision score of 92.69% indicates that the model has a low false positive rate for freshness classification.

Recall: A recall score of 91.05% suggests that the model successfully detects most fresh, dried, or spoiled samples.

F1 Score: A score of 91.15% indicates that the model strikes a good balance between precision and recall, ensuring both high sensitivity and specificity.

2.1.8 Data Augmentation and Model Output

Figure 8 illustrates two key stages in the model workflow: data augmentation and sample predictions.

In Figure 8a, the images of the leaves are augmented by applying various transformations such as rotations, scaling, and flips to reduce over fitting and improving performance on unseen data. The Figure 8b shows the sample predictions of data for each category from the generated model

2.2 Prototype Performance Review

a) Fresh Detection

b) Dried Detection

a) Spoiled Detection

b) Unknown Detection

Figure 9: : Detection Result of Mint leaves for various categories on LCD

2.2.1 *ESP32 Output*

Figure 9 demonstrates the real-time detection results displayed on the ESP32S3-EYE's onboard LCD screen for various categories of mint leaf freshness.

2.2.2 *Serial Port Output*

Figure 10 shows the serial port output from the ESP32S3-EYE, which logs the classification scores for each freshness category.

```
fresh score: 0%, dried score: 0%, spoiled score: 0%, unknown score: 0%
fresh score: 43%, dried score: 12%, spoiled score: 12%, unknown score: 12%
fresh score: 2%, dried score: 92%, spoiled score: 5%, unknown score: 5%
fresh score: 2%, dried score: 93%, spoiled score: 5%, unknown score: 5%
fresh score: 7%, dried score: 79%, spoiled score: 13%, unknown score: 13%
fresh score: 11%, dried score: 71%, spoiled score: 15%, unknown score: 15%
fresh score: 8%, dried score: 67%, spoiled score: 11%, unknown score: 11%
fresh score: 10%, dried score: 60%, spoiled score: 10%, unknown score: 10%
fresh score: 0%, dried score: 0%, spoiled score: 0%, unknown score: 0%
```

Figure 10: Serial port output

2.2.3 *Problems faced*

Several challenges were encountered during the implementation and testing of the freshness classification system, as highlighted below:

Machine Learning Model: A simpler CNN architecture was chosen to ensure compatibility with the hardware limitations of the ESP32S3-EYE. However, this simplicity came at the cost of reduced model accuracy. Dataset: The dataset used exhibited some limitations, particularly in the Dried and Spoiled categories, where a higher rate of false positives was observed. A more extensive and varied dataset would be required to improve model robustness and reduce misclassification.

Indoor Detection: The model was trained and tested in controlled indoor environments. While this allowed for consistent conditions, the model may not generalize well to outdoor environments with more variable lighting and weather conditions. This constraint reduces the overall versatility of the system.

Only One Point Detection: Due to the limited Field of View (FOV) of the ESP32S3-EYE camera, the detection is restricted to a single point or small region of the plant. As a result, the model can only classify a portion of the plant, potentially missing freshness variations across the entire plant.

2.3 Foreword Prospect

Despite the challenges, the project offers significant potential for future improvements and applications, including:

Expansion to Other Crops: The current system can be extended to classify the freshness of other crops, fruits, vegetables, and perishable food items.

Household Applications for Zero Wastage: The system could be adapted for use in household appliances, such as refrigerators, to monitor the freshness of stored food and reduce waste by providing early warnings about spoilage.

Improved Machine Learning Models: Future iterations could incorporate more advanced machine learning architectures, such as ResNet or EfficientNet, to improve accuracy and robustness.

3 Conclusions

This project successfully demonstrates the potential of deploying a lightweight Convolutional Neural Network (CNN) on the ESP32S3-EYE module for real-time freshness classification of mint leaves. The system achieved over 90% accuracy in detecting four categories—Fresh, Dried, Spoiled, and Unknown—while providing live feedback on an integrated display. Despite challenges like dataset limitations, hardware constraints, and simplified model architecture, the system showcases the feasibility of using low-cost IoT devices for agricultural monitoring. Looking ahead, there are opportunities to expand the application to other crops and household appliances, improve the model's accuracy with more advanced architectures, and adapt the system for field-based environments, ultimately promoting better food quality management and supporting sustainable agriculture practices.

4 References

[1] Y. C. J. Liu, S. Yang and Z. Song. Plant leaf classification based on deep learning. In *Chinese Automation Congress (CAC)*, pages 3165–3169. IEEE, 2018.

[2] U. Nation. The 2030 agenda for sustainable development's 17 sustainable development goals (sdgs).

[3] Y. B. K. P. P. C. Rohini Jadhav, Yogesh Suryawanshi. Mint leaves: Dried, fresh, and spoiled dataset for condition analysis and machine learning applications. *Data in Brief*, 51(109717):2352–3409, 2023.

[4] V. Tanwar and S. Lamba. Tea leaf diseases classification and detection using a convolutional neural network. In *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pages 498–502. Ieee, 2023.

[5] X. C. Yue Yuan. Vegetable and fruit freshness detection based on deep features and principal component analysis. *Current Research in Food Science*, 8:2665–9271, 2024.